

# LiDAR Processing

---

1. **LiDAR Sensor**
2. **LiDAR Filtering**
3. **LiDAR Merging**
4. **Obstacle Detection**

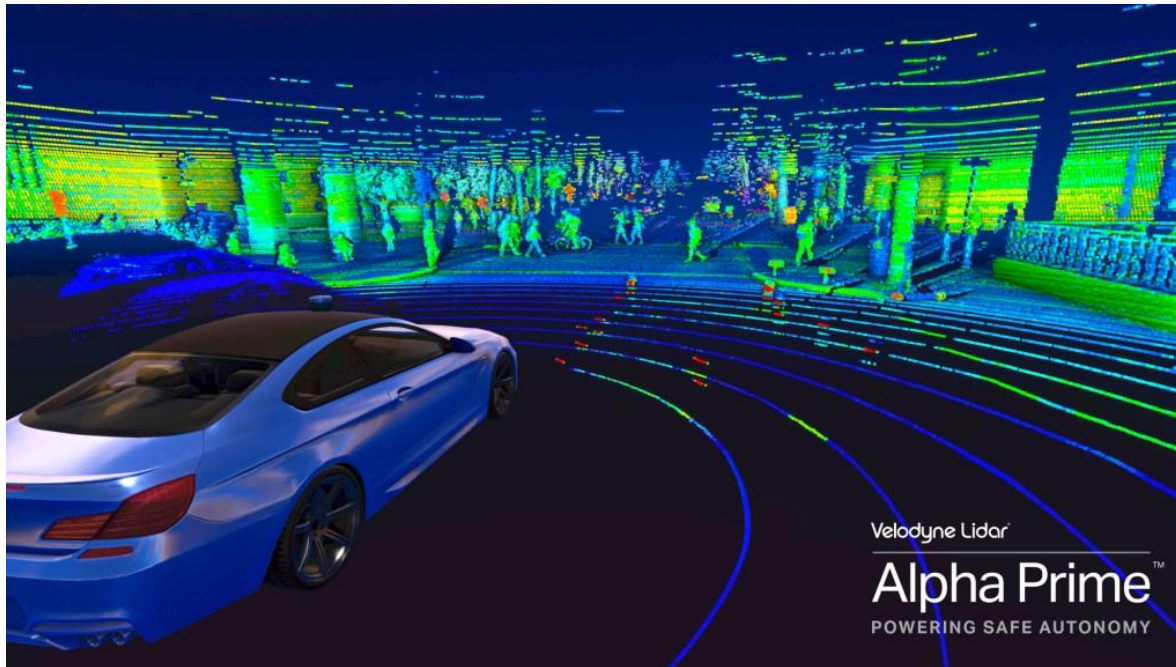
# 01

---

LiDAR Sensor

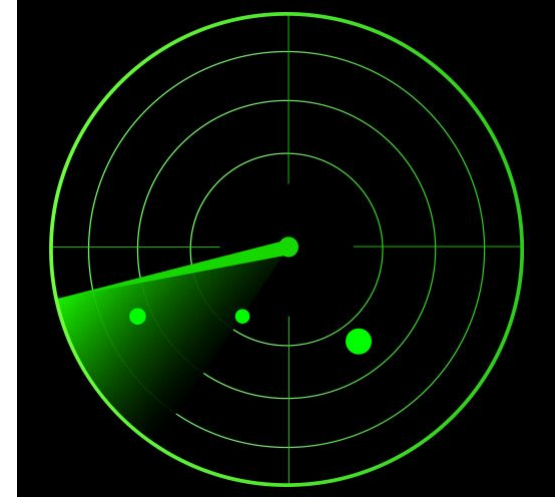
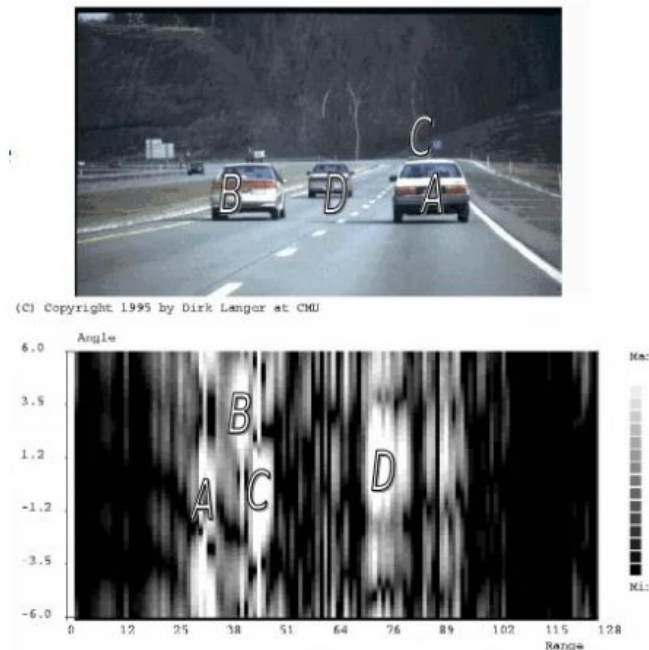
### LiDAR

- Lidar는 레이저를 이용하여, 목표 물체까지의 거리, 방향, 반사율 등을 측정할 수 있는 센서를 의미한다.
- Radar나 Camera에 비해, 물체까지의 거리와 방향에 대한 정보가 매우 우수
- 반면, 날씨에 대한 의존도가 높으며 눈, 비가 오는 날씨에는 사용하기가 힘들



## LiDAR & Radar

- Radar는 전자파 기반의 센서
- 물체의 형상을 확실하게 인식 불가(여기쯤 있겠구나...)
- 기후에 상관없이 항상 제 성능을 발휘
- 사용 사례: 자동차긴급제동장치, 스마트 크루즈 컨트롤, 전방충돌 방지보조 등

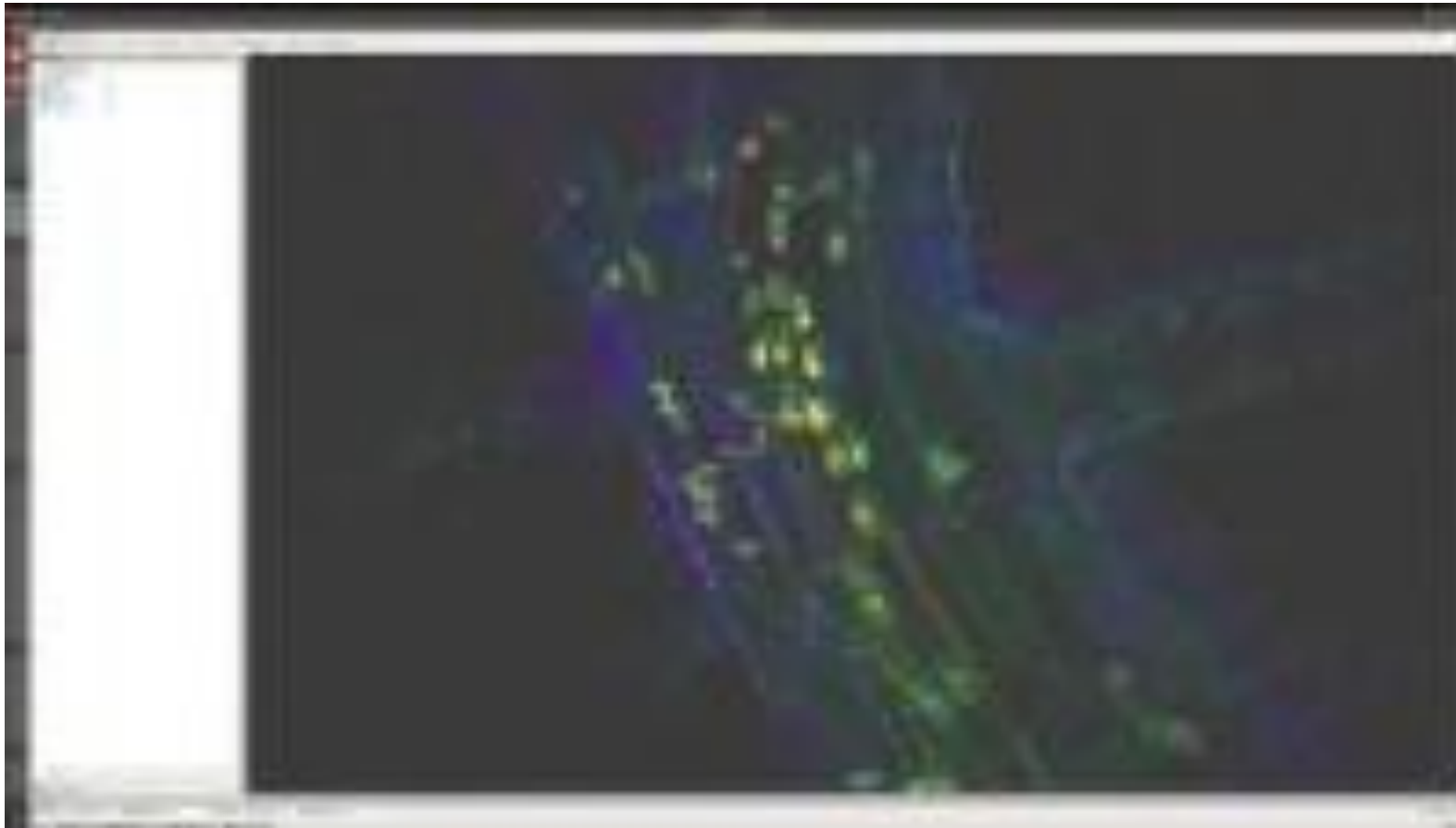


### LiDAR & Radar

구분	발사체	물체 정확도	기후 영향
Radar	전자파	근사치	없음
Lidar	레이저(광선)	거의 정확	있음

## 01. LiDAR

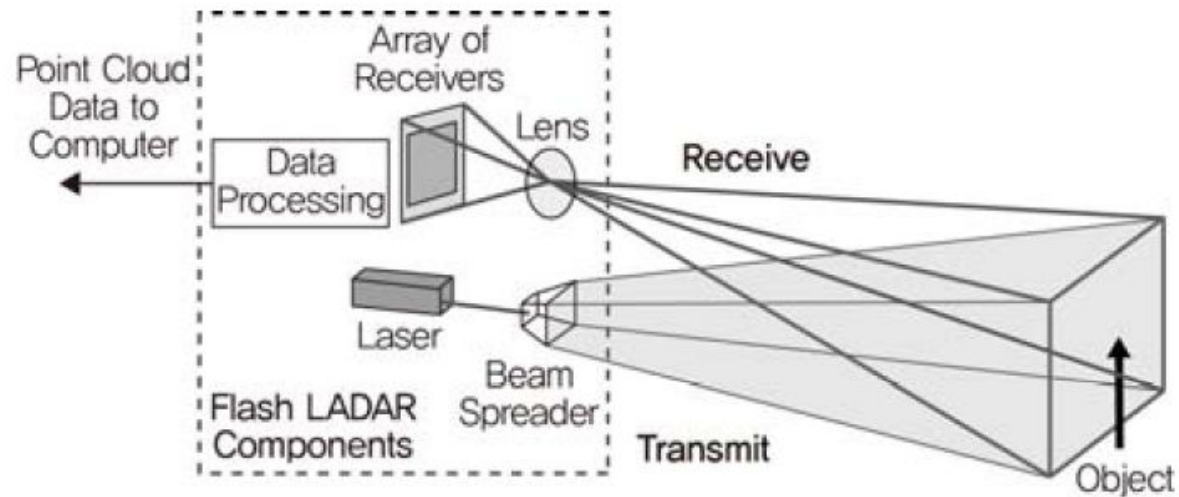
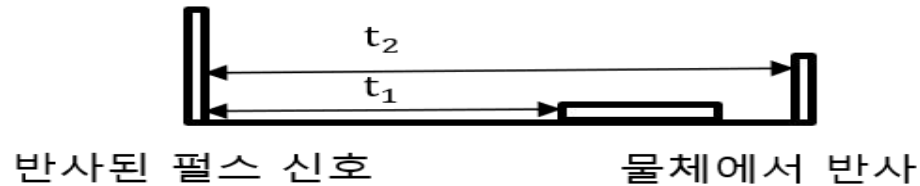
### LiDAR 실차 적용 예시



## 01. LiDAR

### • 작동원리

펄스폭이 매우 짧은 레이저광을 표적으로 보내고, 표적 표면에서 반사되어 되돌아 올 때 까지 시간을 측정하여 표적거리를 이 시간 값과 빛의 속도로부터 산출하는 방법이다. 일반적으로 출력되는 형태는 PointCloud 형태로 출력되며, 다수의 점들이 모여있는 형태이다.





# 02

---

LiDAR Filtering

## 02. LiDAR Filtering

- LiDAR 데이터를 받는 상황에 따라, **Filtering**이 필요
- 특정 거리 이상의 값의 신뢰도가 떨어지는 경우, 최대 출력에 해당하는 **Maximum Range** 값을 조정
- LiDAR의 설치에 따라, 가려지는 각도가 일정 부분 존재할 경우, 그 부분의 데이터를 **Filtering**하여 데이터가 나오지 않도록 함 → **SLAM**을 진행하는 입장에서 매우 중요
- LiDAR의 **Intensity**값에 따라 **Filtering**이 필요할 경우, **Intensity** 기반의 **Filtering**이 가능
- **Polar Coordinate**가 아닌, **Cartesian Coordinate**에서의 **Filtering**이 필요할 때 **Filtering**이 가능

## 02. LiDAR Filtering

- **ROS Laser Filter Package** ([http://wiki.ros.org/laser\\_filters](http://wiki.ros.org/laser_filters))
- LiDAR 데이터를 입력으로하여 이를 필터링하는 패키지
- **Subscriber Topics**
  - /scan - incoming laser scan to filter
- **Published Topics**
  - scan\_filtered - outgoing filtered laser scan

## 02. LiDAR Filtering

- ROS Laser Filter Package ([http://wiki.ros.org/laser\\_filters](http://wiki.ros.org/laser_filters))
- Example Launch File

```
<launch> Sebastian Pütz, 6 years ago • added cartesian box filter
<node pkg="laser_filters" type="scan_to_scan_filter_chain" output="screen" name="laser_filter">
  <remap from="scan" to="base_scan" />
  <rosparm command="load" file="$(find laser_filters)/examples/box_filter.yaml" />
</node>
</launch>
```

- Example YAML File

```
scan_filter_chain:
- name: box_filter
  type: laser_filters/LaserScanBoxFilter
  params:
    box_frame: base_link
    max_x: 0.39
    max_y: 0.34
    max_z: 0.3
    min_x: -0.45
    min_y: -0.34
    min_z: -0.28
    invert: false jailander, 2 years
```

## 02. LiDAR Filtering

- **ROS Laser Filter Package** ([http://wiki.ros.org/laser\\_filters](http://wiki.ros.org/laser_filters))
- 여러 개의 **Filter**를 중첩해서 사용할 수 있으며, 출력을 **LaserScan** 메시지 타입이 아닌, **PointCloud** 메시지 타입으로 출력도 가능하다.
- 사용 가능한 **Filter**로는 `box_filter`, `footprint_filter`, `intensity_filter`, `mask_filter`, `median_filter`, `multiple_filter`, `pass_through`, `polygon_filter`, `range_filter`, `shadow_filter`, `speckle_filter` 가 있다.

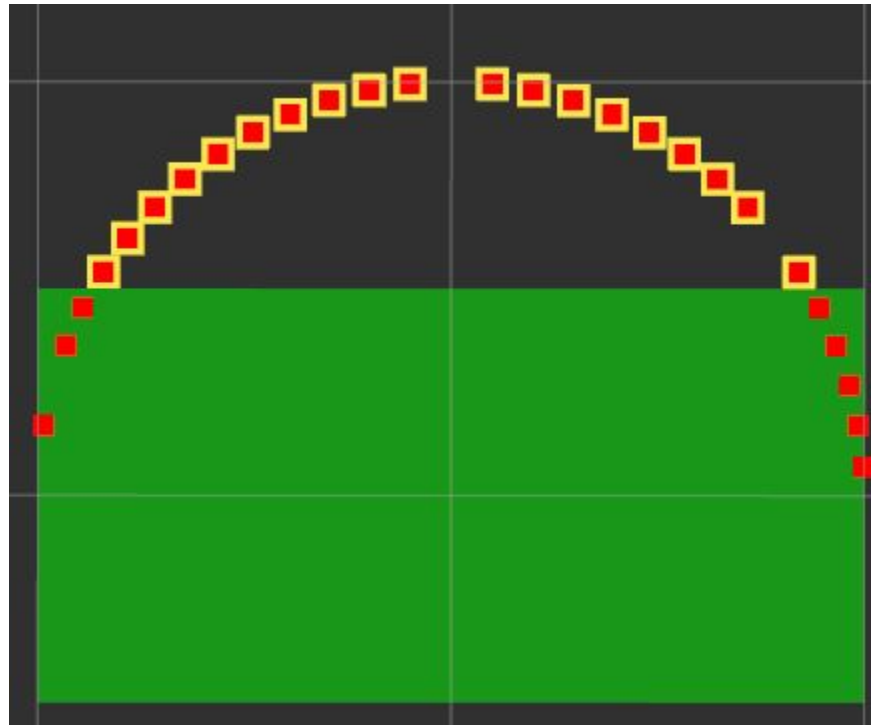
## 02. LiDAR Filtering

- ROS Laser Filter Package ([http://wiki.ros.org/laser\\_filters](http://wiki.ros.org/laser_filters))
- box filter
- Cartesian Coordinate에서 박스 형태로 라이다 값을 필터링

```
scan_filter_chain:  
- name: box_filter  
  type: laser_filters/LaserScanBoxFilter  
  params:  
    box_frame: laser  
    max_x: 0.5  
    max_y: 1.0  
    max_z: 50.0  
    min_x: -0.5  
    min_y: -1.0  
    min_z: -50.0  
    inverted : false
```

## 02. LiDAR Filtering

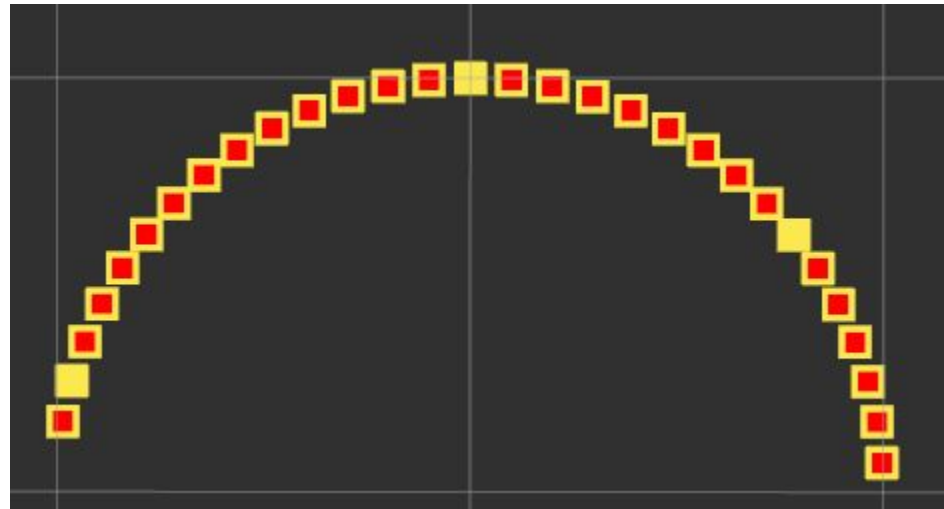
- ROS Laser Filter Package ([http://wiki.ros.org/laser\\_filters](http://wiki.ros.org/laser_filters))
- box filter
- Cartesian Coordinate에서 박스 형태로 라이다 값을 필터링
- 



## 02. LiDAR Filtering

- ROS Laser Filter Package ([http://wiki.ros.org/laser\\_filters](http://wiki.ros.org/laser_filters))
- Interpolation Filter
- 중간 중간의 빈 부분을 채워주는 Filter

```
scan_filter_chain:  
- name: interpolation_filter  
  type: laser_filters/InterpolationFilter
```



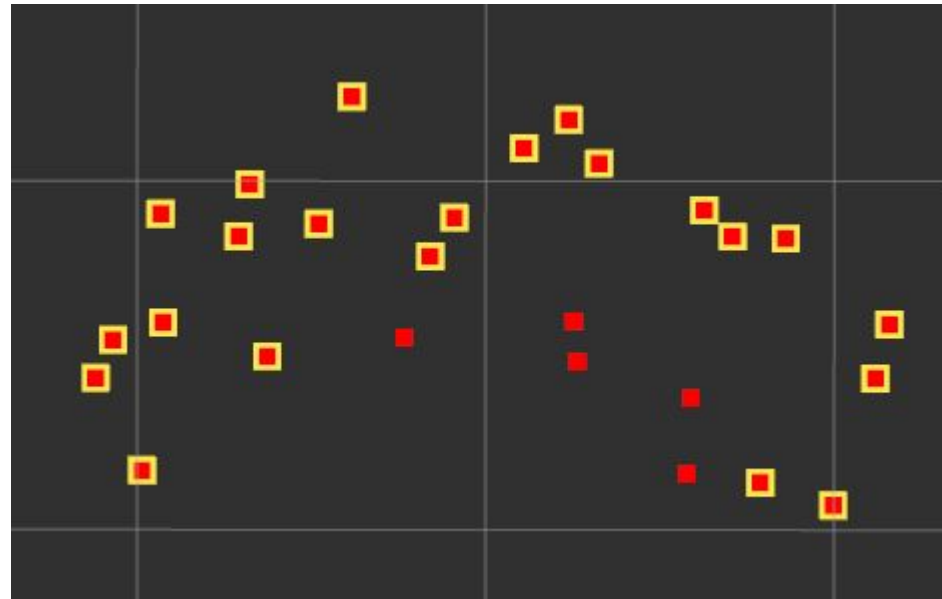


## 02. LiDAR Filtering

- ROS Laser Filter Package ([http://wiki.ros.org/laser\\_filters](http://wiki.ros.org/laser_filters))
- LaserScanIntensityFilter
- Laser Scan의 Intensity를 이용하여, Filtering을 진행
- LiDAR의 종류에 따라 Intensity값이 출력되는 센서의 경우, 위 필터를 이용하여 필터링에 사용할 수 있음

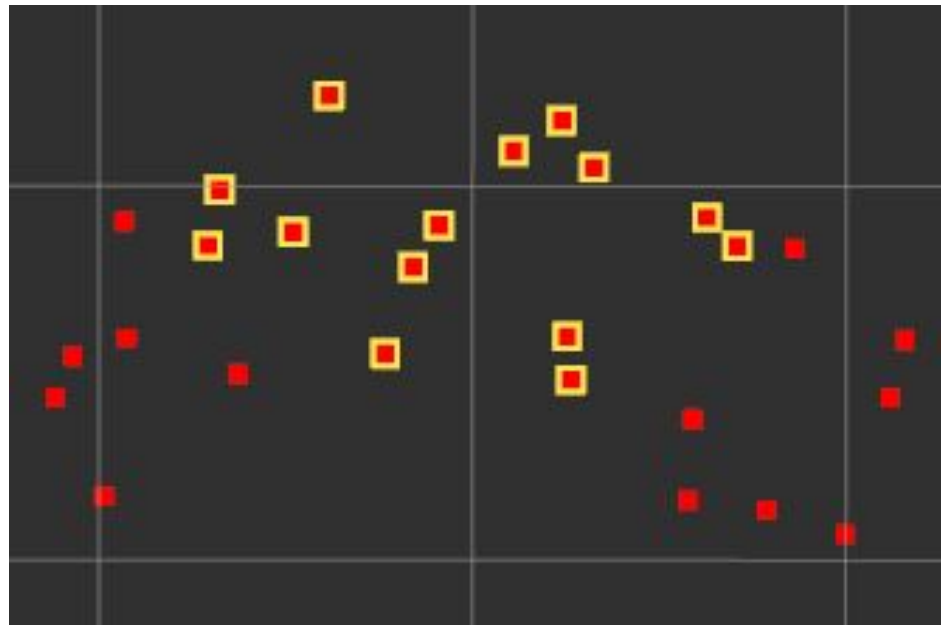
## 02. LiDAR Filtering

- ROS Laser Filter Package ([http://wiki.ros.org/laser\\_filters](http://wiki.ros.org/laser_filters))
- LaserScanRangeFilter
- Laser Scan의 Range를 이용하여, Filtering을 진행



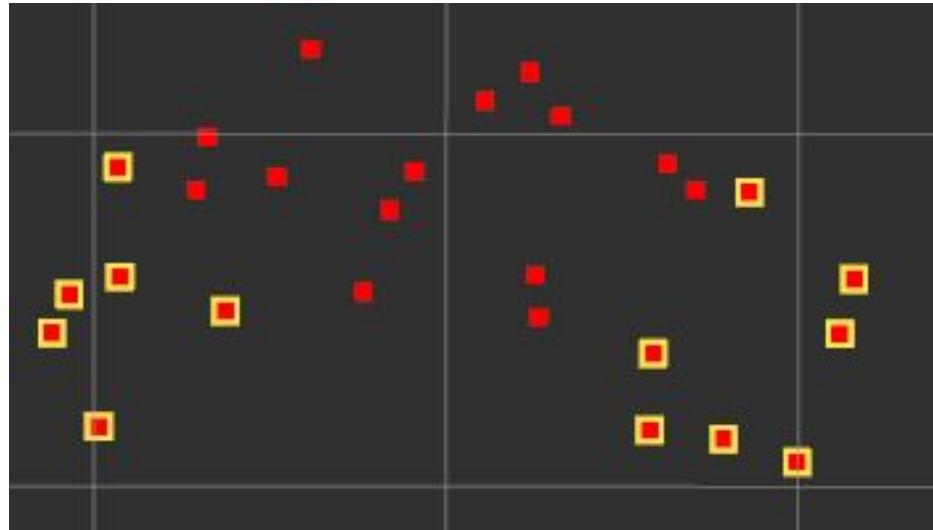
## 02. LiDAR Filtering

- ROS Laser Filter Package ([http://wiki.ros.org/laser\\_filters](http://wiki.ros.org/laser_filters))
- LaserScanAngularBoundsFilter
- Laser Scan의 측정하는 Angle을 이용하여, Filtering을 진행
- 정해진 Angle 범위를 벗어나는 데이터를 필터링



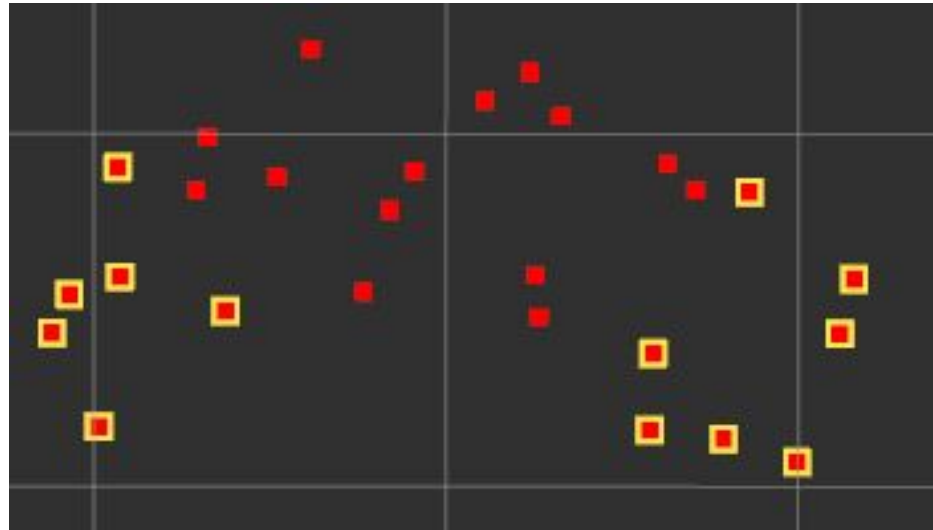
## 02. LiDAR Filtering

- ROS Laser Filter Package ([http://wiki.ros.org/laser\\_filters](http://wiki.ros.org/laser_filters))
- LaserScanAngularBoundsFilterInPlace
- Laser Scan의 측정하는 Angle을 이용하여, Filtering을 진행
- 정해진 Angle 범위 내부의 데이터를 필터링



## 02. LiDAR Filtering

- ROS Laser Filter Package ([http://wiki.ros.org/laser\\_filters](http://wiki.ros.org/laser_filters))
- LaserScanAngularBoundsFilterInPlace
- Laser Scan의 측정하는 Angle을 이용하여, Filtering을 진행
- 정해진 Angle 범위 내부의 데이터를 필터링



# 03

---

LiDAR Merging

### 03. LiDAR Merging

- **ROS ira\_laser\_tools Package ([http://wiki.ros.org/ira\\_laser\\_tools](http://wiki.ros.org/ira_laser_tools))**
- 다수의 LaserScan 데이터를 입력으로 받아서, 이를 하나의 LaserScan 메시지로 전달하는 패키지
- laserscan\_multi\_merger와 laserscan\_virtualizer로 구성된다.
- rqt\_reconfigure를 통해 실시간으로 Parameter를 수정할 수 있다.

### 03. LiDAR Merging

- **ROS ira\_laser\_tools Package ([http://wiki.ros.org/ira\\_laser\\_tools](http://wiki.ros.org/ira_laser_tools))**
- 다수의 Laser Scan을 하나의 Frame으로 통합하는 역할을 진행
- 각 라이다 사이의 거리 및 위치를 알아야만, 이에 대한 **Frame** 결합이 가능
- TF 패키지에 있는 **Static\_Transform\_Publisher**를 이용하여, **Frame** 간 결합이 가능
- Rviz를 통해, 결합된 **Frame**에서 두 개의 **LaserScan**이 정상적으로 나오는 상태에서 동작
- 일반적인 **SLAM** 알고리즘에서 **LiDAR** 데이터를 1개만 사용하므로, 위 패키지를 이용하여 통합



# 04

---

## Obstacle Detection

## 04. Obstacle Detection

### - LiDAR Clustering

- LiDAR의 경우, 공간에 대한 Resolution이 매우 높은편이므로, 하나의 물체에서 다수의 측정값이 발생할 수 있음
- 그러므로, 발생하는 다수의 측정값을 묶어서, 각 물체별로 분류하는 Clustering이 필요



## 04. Obstacle Detection

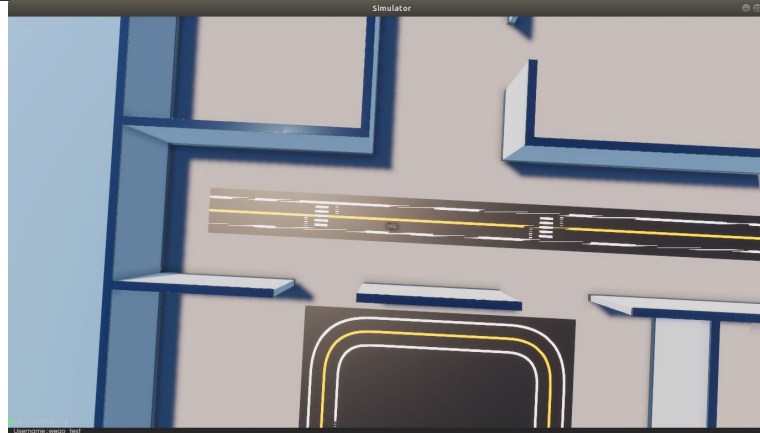
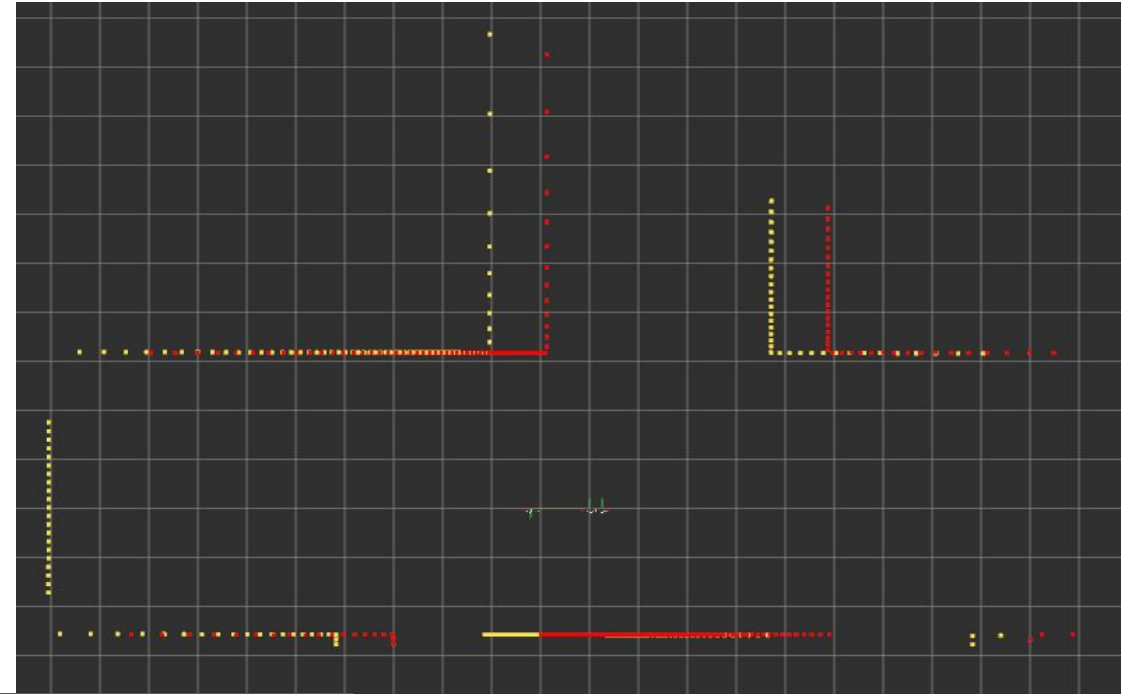
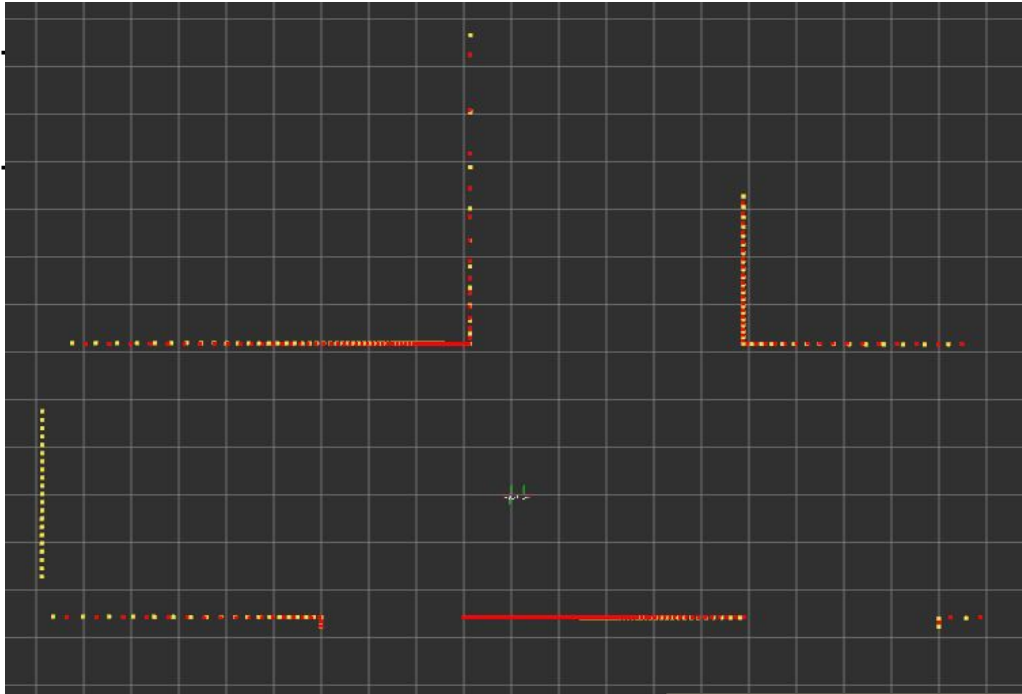
- **Obstacle Detection**
- 일반적으로 진행하는 Obstacle Detection의 과정은 다음과 같다.
- 1. LiDAR Calibration
- 2. LiDAR Filtering
- 3. LiDAR Clustering
- 4. Box or Circle Fitting
- 5. Obstacle Tracking

## 04. Obstacle Detection

- **LiDAR Calibration**
- LiDAR Calibration은 다수의 라이다를 사용할 경우, 하나의 **Frame**으로의 통합을 위한 역할을 수행한다.
- 좌표계를 통합하여, 하나의 좌표계에서 다수의 라이다를 정확하게 확인하기 위한 과정
- 여러 개의 물체를 두고, 하나의 좌표계로 통합했을 때, 같은 물체가 정확하게 잡히는지 확인이 필요

## 04. Obstacle Detection

### - LiDAR Calibration



## 04. Obstacle Detection

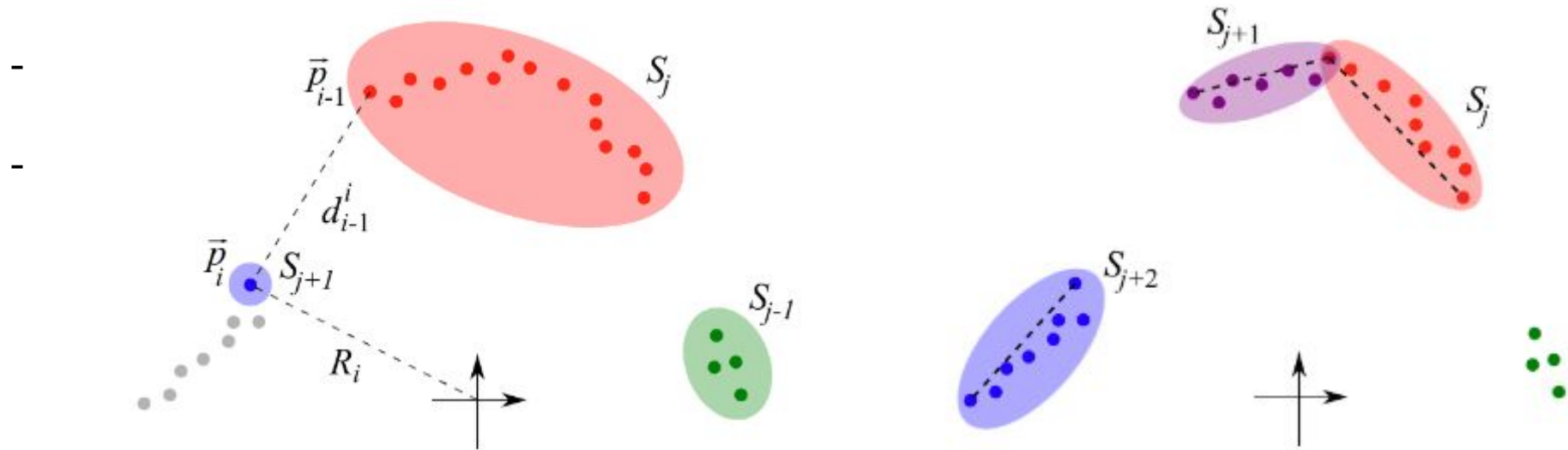
- LiDAR Filtering
- laser\_filter 등을 이용하여 LiDAR Data를 Filtering 진행
-

## 04. Obstacle Detection

### - LiDAR Clustering

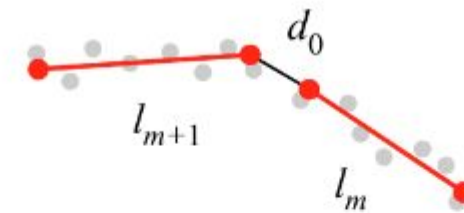
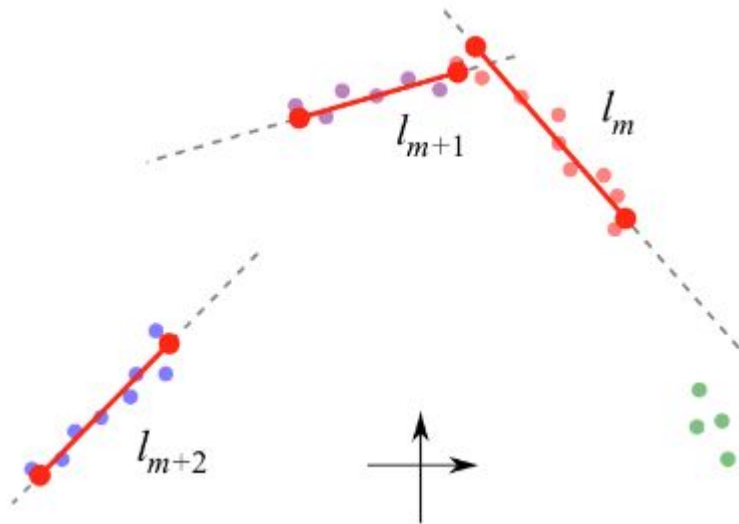
- Calibration 및 Filtering, Merging이 끝난 LiDAR 데이터를 이용하여, 의미있는 물체를 뽑아내는 과정

- 과정으로는 Grouping Split Merge 등으로 이루어져 있다

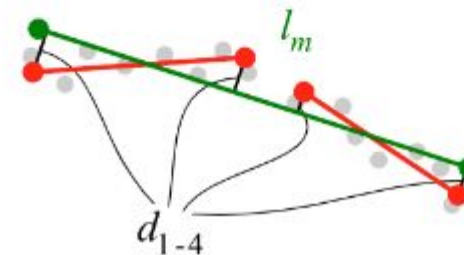


## 04. Obstacle Detection

- LiDAR Clustering
- Calibration 및 Filtering, Merging이 끝난 LiDAR 데이터를 이용하여, 의미있는 물체를 뽑아내는 과정
- 과정으로는 Grouping, Split, Merge 등으로 이루어



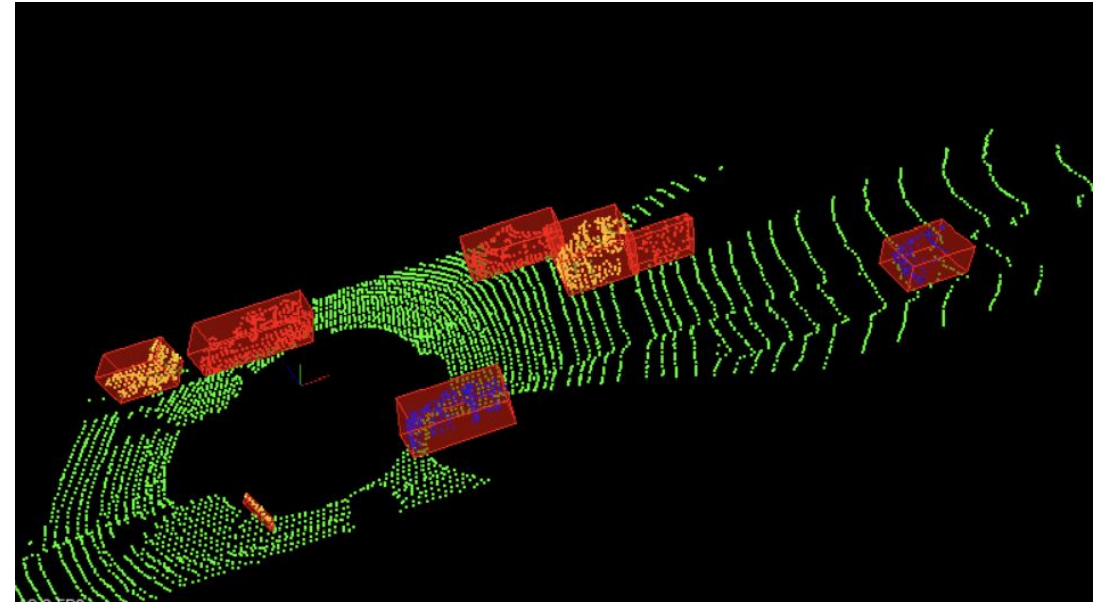
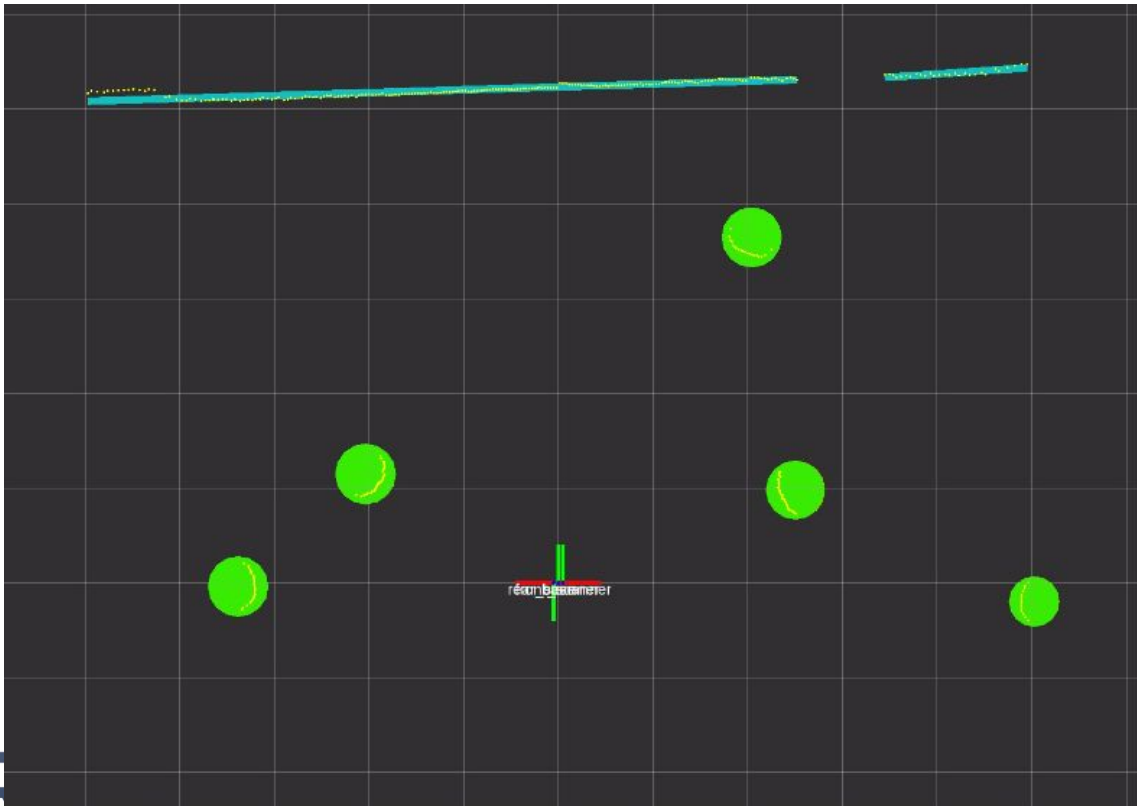
**Figure 5:** Graphic representation of segments merging criterion: connectivity test.





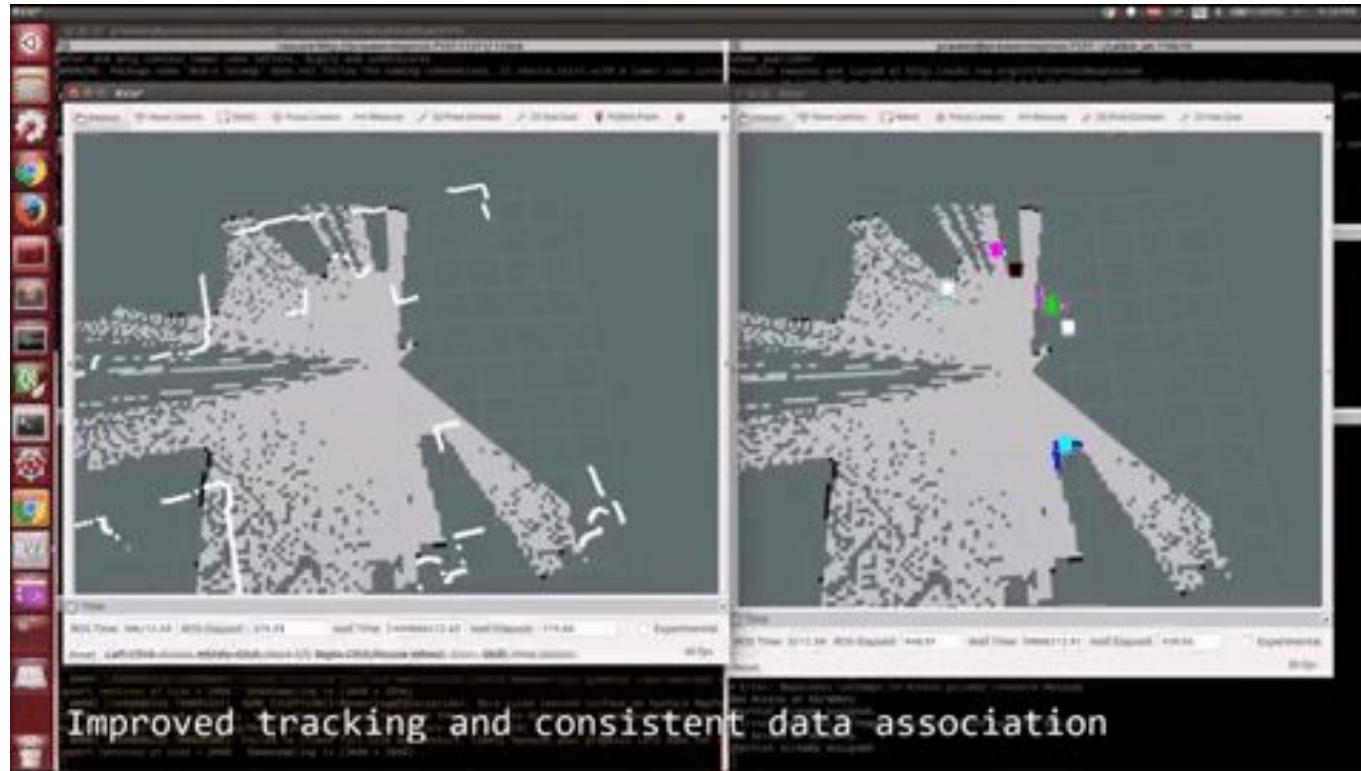
## 04. Obstacle Detection

- Circle or Box Fitting
- Clustering을 통해 나온 결과를 바탕으로, 물체를 원형 또는 사각형 박스 형태로 Fitting하는 기법
- 



## 04. Obstacle Detection

- **Obstacle Tracking**
- 물체로 판별된 Obstacle을 Kalman Filter 등을 이용하여, Tracking하는 과정
- Multi-Object Tracking (MoT)





Q & A

[go.support@wego-robotics.com](mailto:go.support@wego-robotics.com)

[go.sales@wego-robotics.com](mailto:go.sales@wego-robotics.com)